

03 . M a y . 2022

# Non-Random Circular Jittering Algorithm using Trigonometric Functions for Scatter Plots of Ordinal Outcomes

PHUSE 2022

***REGENERON***<sup>®</sup>

# Agenda

## 1. Introduction

- Current Approaches and Limitations

## 2. Non-Random Jittering Method

- Dataset preparation
- Applying non-random circular jittering algorithm
- Create the actual scatter plot using the jittered points

## 3. Example

# Agenda

## 1. Introduction

- Current Approaches and Limitations

## 2. Non-Random Jittering Method

- Dataset preparation
- Applying non-random circular jittering algorithm
- Create the actual scatter plot using the jittered points

## 3. Example

# Introduction

- Scatter plots are commonly used in visualizing the relationship between continuous outcomes
- Ordinal outcomes due to limited range of the scale often results in numerous overlapping points
- Commonly used visualization methods have various shortcomings:
  - Bubble Plot, Use of Transparency: Not intuitive in visualizing the degree of overlap
  - Beehive, Boxplot: Losing the ordinal nature of the outcome being plotted
  - Random Jittering: Erratic displays relying on randomness for the placement of each point

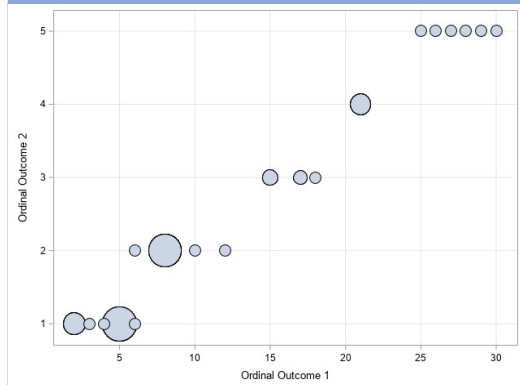
Therefore, there is a need to develop an alternative method  
without the above shortcomings

A novel non-random circular jittering  
algorithm was developed to meet this need

# Current Approaches and Limitations

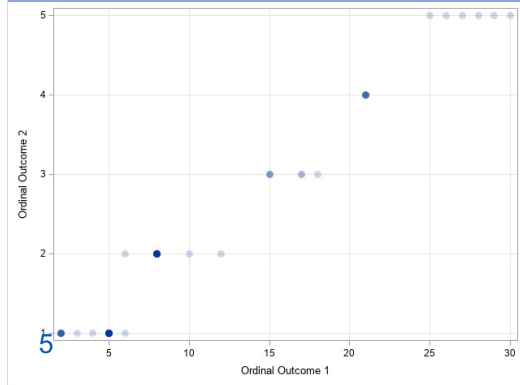
## Bubble Plot

Not intuitive in visualizing the degree of overlap



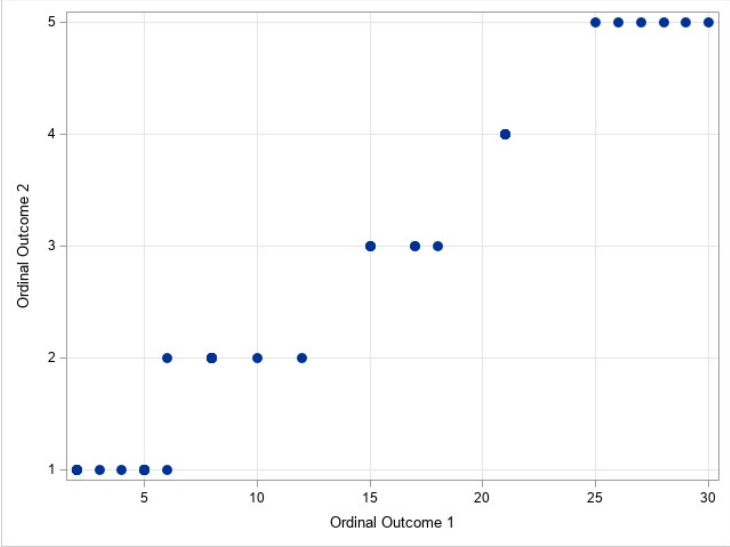
## Transparency

Not intuitive in visualizing the degree of overlap



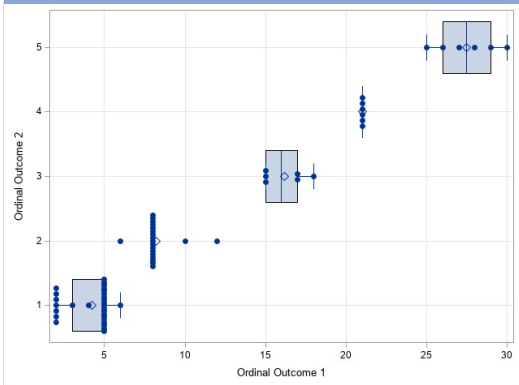
## Basic Scatter Plot

Unable to distinguish overlap



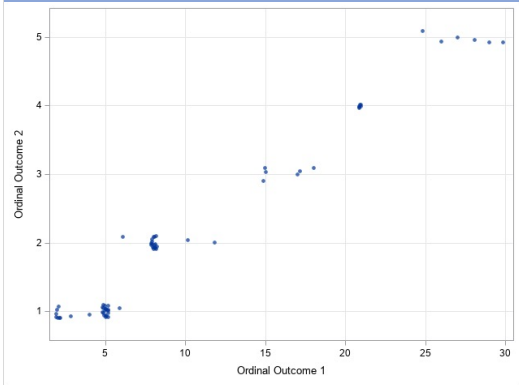
## Beehive/Boxplot

Loss of the ordinal nature of the outcome being plotted



## Random Jittering

Random placement of points



# Agenda

## 1. Introduction

- Current Approaches and Limitations

## 2. Non-Random Jittering Method

- Dataset preparation
- Applying non-random circular jittering algorithm
- Create the actual scatter plot using the jittered points

## 3. Example

# Novel Algorithm: Non-Random Circular Jittering

## Main Concept:

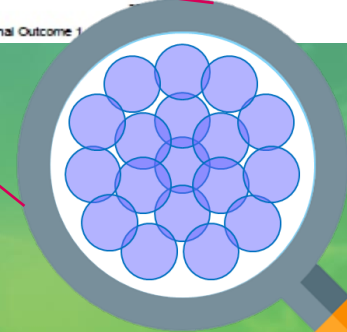
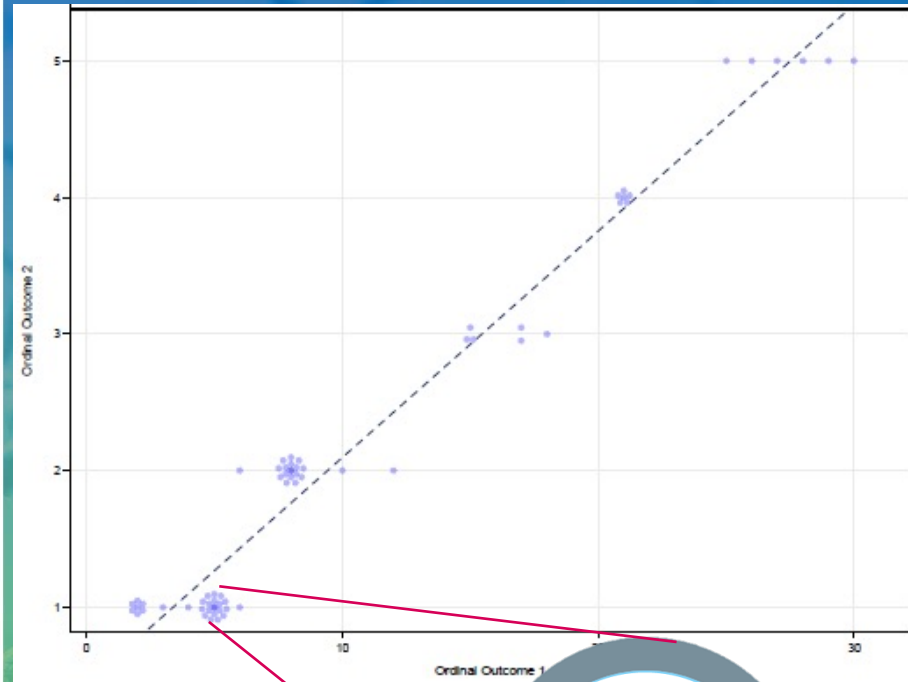
Jitter points around concentric circles based on the original x-y coordinate

## Advantage:

Maintain scatter plot integrity as all data points are plotted and retains the ordinal scale being plotted resulting in predictable plots centered around the original observation

## General Steps:

- Calculate perimeter of concentric circles
- Place points evenly around the concentric circle
- Once the circle is full, then create a larger concentric circle and repeat



# Non-Random Jittering Method

## 3 Main Steps:

- 1) Dataset preparation
- 2) Applying non-random circular jittering algorithm
  - a) Rescale to ensure that jittering results in a circle
  - b) Handling the first 4 points
  - c) Apply non-random circular jittering algorithm
- 3) Create the actual scatter plot using the jittered points

# Non-Random Jittering Method

## Step 1: Dataset Preparation

- In example below, Endpoint-X and Endpoint-Y are ordinal endpoints. Table 1 is the input dataset; Table 2 includes derived variables
- Calculate the degree of overlap for each observed x-y coordinate which is stored in the variable  $CT_{overall}$
- Each point is assigned a unique sequence number stored in the variable SEQ for each x-y coordinate

Table 1: Input Dataset

| Observation ID | EndpointX | EndpointY |
|----------------|-----------|-----------|
| 1              | 2         | 1         |
| 2              | 2         | 1         |
| 3              | 2         | 1         |
| 4              | 2         | 1         |
| 5              | 2         | 1         |
| 6              | 2         | 1         |
| 7              | 2         | 1         |
| 8              | 3         | 1         |
| 9              | 4         | 1         |
| 10             | 5         | 1         |
| 11             | 5         | 1         |
| 12             | 5         | 1         |
| 13             | 5         | 1         |
| 14             | 5         | 1         |
| 15             | 5         | 1         |
| 16             | 5         | 1         |
| 17             | 5         | 1         |
| 18             | 5         | 1         |
| 19             | 5         | 1         |
| 9              | ...       | ...       |

Table 2: Pre-Jittering Processed Dataset

| Observation ID | EndpointX | EndpointY | CT <sub>overall</sub> | SEQ <sub>overall</sub> | Circle i | n i | SEQ <sub>ij</sub> |
|----------------|-----------|-----------|-----------------------|------------------------|----------|-----|-------------------|
| 1              | 2         | 1         | 7                     | 1                      | 0        | 1   | 1                 |
| 2              | 2         | 1         | 7                     | 2                      | 1        | 6   | 1                 |
| 3              | 2         | 1         | 7                     | 3                      | 1        | 6   | 2                 |
| 4              | 2         | 1         | 7                     | 4                      | 1        | 6   | 3                 |
| 5              | 2         | 1         | 7                     | 5                      | 1        | 6   | 4                 |
| 6              | 2         | 1         | 7                     | 6                      | 1        | 6   | 5                 |
| 7              | 2         | 1         | 7                     | 7                      | 1        | 6   | 6                 |
| 8              | 3         | 1         | 1                     | 1                      | 0        | 1   | 1                 |
| 9              | 4         | 1         | 1                     | 1                      | 0        | 1   | 1                 |
| 10             | 5         | 1         | 10                    | 1                      | 0        | 1   | 1                 |
| 11             | 5         | 1         | 10                    | 2                      | 1        | 6   | 1                 |
| 12             | 5         | 1         | 10                    | 3                      | 1        | 6   | 2                 |
| 13             | 5         | 1         | 10                    | 4                      | 1        | 6   | 3                 |
| 14             | 5         | 1         | 10                    | 5                      | 1        | 6   | 4                 |
| 15             | 5         | 1         | 10                    | 6                      | 1        | 6   | 5                 |
| 16             | 5         | 1         | 10                    | 7                      | 1        | 6   | 6                 |
| 17             | 5         | 1         | 10                    | 8                      | 2        | 3   | 1                 |
| 18             | 5         | 1         | 10                    | 9                      | 2        | 3   | 2                 |
| 19             | 5         | 1         | 10                    | 10                     | 2        | 3   | 3                 |
| ...            | ...       | ...       | ...                   | ...                    | ...      | ... | ...               |

# Non-Random Jittering Method

## Step 2a: Rescaling to ensure circular jittering

- Adjust the x- and y-axes if they are on different scales to ensure that the jittering results in a circle as opposed to an oval.
- For example:      Endpoint-X is on a scale from 0 to 30 while  
                                 Endpoint-Y is on a scale from 0 to 5
- The adjustment is achieved by calculating the x-y ratio as follows:

$$ratio_{xy} = \frac{EndpointX\ range}{EndpointY\ range}$$

# Non-Random Jittering Method

## Step 2b: Handling the first four overlapping points

First 4 points are handled somewhat differently

- 1) If only **one** point, point is not jittered and the original coordinates are retained
- 2) If there are **two** overlapping points, jitter the points vertically
- 3) If **three** overlapping points, jitter the points into an equilateral triangle
- 4) If there are **four** points, one point at the original coordinate, and the remainder 3 points will form an equilateral triangle



# Non-Random Jittering Method

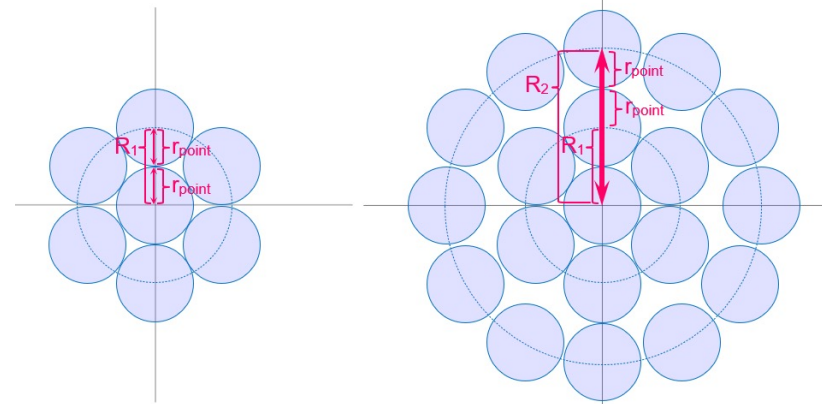
## Step 2c: Non-Random Circular Jittering Concept

- After 4<sup>th</sup> overlapping point, non-random circular jittering algorithm begins
- Underlying concept:
  - 1) Concentric circles around the original x-y coordinate are drawn
  - 2) Overlapping points are distributed equally spaced placed along the concentric circles until the circle is filled
  - 3) Once the circle is filled, the process continues with a larger concentric circle until all points are placed

# Non-Random Jittering Method

## Step 2c: Non-Random Circular Jittering Implementation

- First step:
  - Calculate perimeter of the circle (where  $i$  represents the concentric circle number, starting with  $i = 1$ )
  - Radius of the circle ( $R_i$ ) is the radius of the previous circle ( $R_{i-1}$ ) (for the first circle the radius of the previous circle would be zero) plus twice the radius of each point ( $r_{\text{point}}$ ):  $R_i = R_{i-1} + 2 * r_{\text{point}}$
  - Perimeter of the  $i^{\text{th}}$  circle is  $2 \times \pi \times R_i$



# Non-Random Jittering Method

## Step 2c: Non-Random Circular Jittering Implementation

- Second step:
  - Determine the number of points that will fit in the first concentric circle
  - Calculation is performed by dividing the perimeter by twice the radius of each point:  $n_i = (2 \times \pi \times R_i) / (2 \times r_{\text{point}})$

# Non-Random Jittering Method

## Step 2c: Non-Random Circular Jittering Implementation

- Third step:
  - Place jittered points along the perimeter of concentric circles:
    - Starting point: Start at (0,1) at 12 o'clock position on the y-axis. Since the origin of a unit circle is at (1,0) at 3 o'clock position, add 90° in the counter-clockwise direction.
    - Evenly distributed placement: Placement of each point will be evenly distributed along the perimeter. Therefore, the 360° will be divided by  $n_i$  which is the number of points to be placed within the circle  $i$ , so that each point denoted as  $SEQ_{i,j}$  will be placed evenly around the circle. As noted above, 90° will be added to start the first point at the (0,1) position.

$$degrees = 90 + \frac{360}{n_i} \times (SEQ_{i,j} - 1)$$

# Non-Random Jittering Method

## Step 2c: Non-Random Circular Jittering Implementation

- If there are 3 points to be placed ( $n_i = 3$ ), then each point will be  $120^\circ$  apart from each other ( $360^\circ / 3$ ) and placed based on the previously provided formula:

$$degrees = 90 + \frac{360}{n_i} \times (SEQ_{i,j} - 1)$$

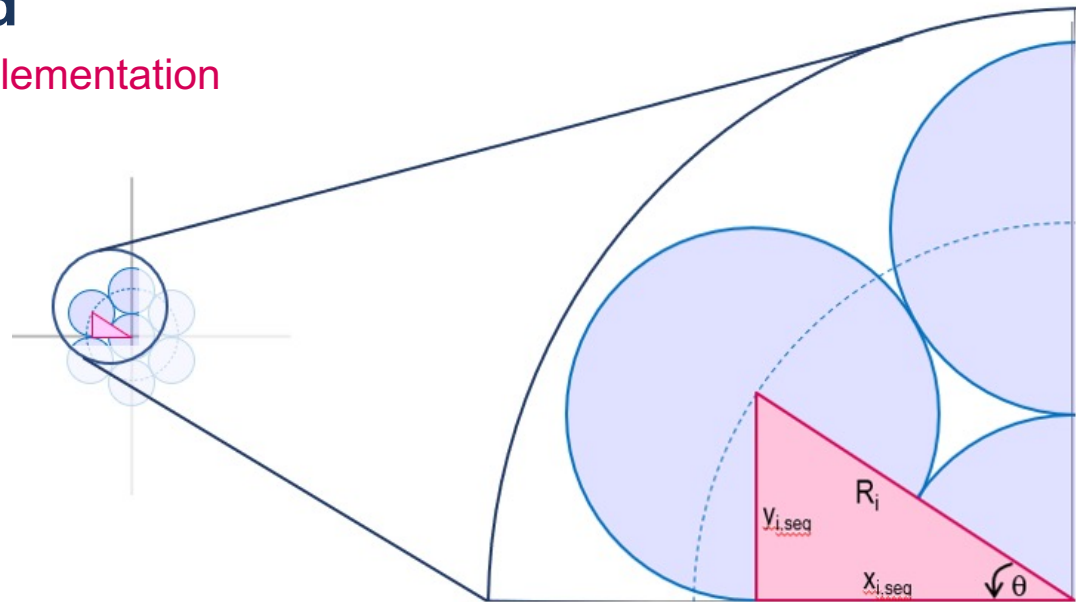
- First point  $SEQ_{1,1} = 1$  will be placed at  $90^\circ$  (since starting point is at  $90^\circ$  as noted earlier)
  - Second point  $SEQ_{1,2} = 2$  will be placed at  $210^\circ$  ( $90^\circ + 120^\circ \times 1$ )
  - Third point  $SEQ_{1,3} = 3$  will be placed at  $330^\circ$  ( $90^\circ + 120^\circ \times 2$ )
- Since the input for the trigonometric functions in SAS is in radians, the following conversion is required:

$$radians = \frac{degrees}{360} \times 2 \times \pi$$

# Non-Random Jittering Method

## Step 2c: Non-Random Circular Jittering Implementation

- Fourth step:
  - Calculate the x-y coordinates
  - Convert degrees into radians
  - Adjust the scale using the ratio<sub>x-y</sub>
  - Use trigonometric functions to calculate the x-y coordinates



$$x_{jitter} = x_{original} + \cos\left(\frac{90 + \frac{360}{n_i} \times (SEQ_{i,j} - 1)}{360} \times 2 \times \pi\right) \times R_i \times ratio_{xy}$$

$$y_{jitter} = y_{original} + \sin\left(\frac{90 + \frac{360}{n_i} \times (SEQ_{i,j} - 1)}{360} \times 2 \times \pi\right) \times R_i$$

# Agenda

## 1. Introduction

- Current Approaches and Limitations

## 2. Non-Random Jittering Method

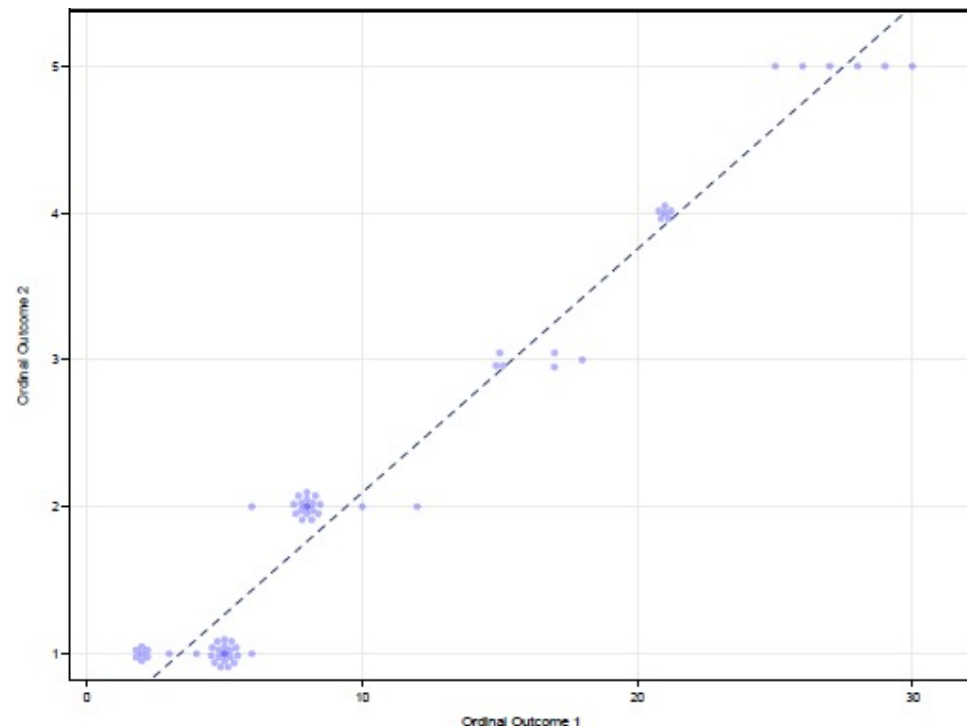
1. Dataset preparation
2. Applying non-random circular jittering algorithm
3. Create the actual scatter plot using the jittered points

## 3. Example

# Example

| Jittered Dataset                 |                |           |           |     |     |                  |
|----------------------------------|----------------|-----------|-----------|-----|-----|------------------|
|                                  | Observation ID | EndpointX | EndpointY | CT  | SEQ |                  |
| 7 overlapping points at (2,1)    | 1              | 2         | 1         | 7   | 1   | EndpointX_Jitter |
|                                  | 2              | 2         | 1         | 7   | 2   | EndpointY_Jitter |
|                                  | 3              | 2         | 1         | 7   | 3   |                  |
|                                  | 4              | 2         | 1         | 7   | 4   |                  |
|                                  | 5              | 2         | 1         | 7   | 5   |                  |
|                                  | 6              | 2         | 1         | 7   | 6   |                  |
|                                  | 7              | 2         | 1         | 7   | 7   |                  |
| 1 non-overlapping point at (3,1) | 8              | 3         | 1         | 1   | 1   |                  |
| 1 non-overlapping point at (4,1) | 9              | 4         | 1         | 1   | 1   |                  |
| 10 overlapping points at (5,1)   | 10             | 5         | 1         | 10  | 1   |                  |
|                                  | 11             | 5         | 1         | 10  | 2   |                  |
|                                  | 12             | 5         | 1         | 10  | 3   |                  |
|                                  | 13             | 5         | 1         | 10  | 4   |                  |
|                                  | 14             | 5         | 1         | 10  | 5   |                  |
|                                  | 15             | 5         | 1         | 10  | 6   |                  |
|                                  | 16             | 5         | 1         | 10  | 7   |                  |
|                                  | 17             | 5         | 1         | 10  | 8   |                  |
|                                  | 18             | 5         | 1         | 10  | 9   |                  |
|                                  | 19             | 5         | 1         | 10  | 10  |                  |
|                                  | ...            | ...       | ...       | ... | ... |                  |

# Non-Random Circular Jittering



- Maintain the scatter plot integrity as all data points are plotted unlike the bubble plot or transparency methods
- Ordinal scale is retained unlike converting one axis to categorical and using the box/beeswarm plot
- Predictable scatter plots centered around the original observation
- Improve scatter plots for ordinal outcomes with considerable amount of overlapping points

# Acknowledgments

- Rui Huang (principal author), Oxford University
- Michael Pannucci (presenting author), Regeneron
- Toshio Kimura, Regeneron

# Q&A